

**ВНЗ Укоопспілки «Полтавський університет економіки
і торгівлі»**

*Кафедра математичного моделювання та соціальної
інформатики*



**КОМП'ЮТЕРНІ НАУКИ І ПРИКЛАДНА
МАТЕМАТИКА
(КНіПМ-2018)**

**МАТЕРІАЛИ НАУКОВО-ПРАКТИЧНОГО
СЕМІНАРУ
Випуск 1**

Полтава
2018

КОМП'ЮТЕРНІ НАУКИ І ПРИКЛАДНА МАТЕМАТИКА (КНіПМ-2018): матеріали науково-практичного семінару. Випуск 1 / за ред. Ємця О.О. – Полтава: Кафедра ММСІ ПУЕТ, 2018. – 64 с.

Збірник матеріалів науково-практичного семінару містить сучасну проблематику в таких галузях інформатики та системних наук, як теоретичні основи інформатики та кібернетики, математичне моделювання та обчислювальні методи, математичне та програмне забезпечення обчислювальних машин і систем, системний аналіз і теорія оптимальних рішень. Представлено тези доповідей, що відображають проблеми сучасної підготовки фахівців з комп'ютерних наук, прикладної математики, системного аналізу та комп'ютерних інформаційних технологій.

*Матеріали друкуються в авторській редакції мовами оригіналів.
За виклад, зміст і достовірність матеріалів відповідають автори*

Ум. друк. арк. 4,0
©Кафедра ММСІ ПУЕТ, 2018

ЗМІСТ

Сивокінь О. Ю., Ємець О. О. Розробка та програмна реалізація тренажера з теми «Метод Ленда та Дойга» дистанційного курсу «Методи оптимізації та дослідження операцій».....	4
Безмєнов Е. Г. Розробка програмного забезпечення для тренажера з теми «Неоднорідні диференціальні рівняння другого порядку із сталими коефіцієнтами» з дисципліни «Математичний аналіз».....	9
Олексійчук Ю. Ф., Голубенко В. О. Програмна реалізація елементів тренажеру з теми «Сортування бульбашками» дисципліни «Аналіз алгоритмів».....	11
Дяченко В.О. Алгоритмізація методу задоволення вимог при прийнятті рішень за умов визначеності.....	17
Куркін В. В., Ємець О. О. Алгоритмізація та програмування елементів тренажера з побудови математичної моделі комбінаторної оптимізаційної “Задачі директора”.....	21
Мандрика В. М., Олексійчук Ю. Ф. Тренажер з теми «1-R алгоритм» дисципліни «Комп’ютерний аналіз статистичних даних».....	27
Стовбун Д. О., Ємець О. О. Алгоритмізація та програмування елементів тренажера з побудови математичної моделі комбінаторної оптимізаційної задачі про оптимізацію суміші.....	31
Шербак О.В. Постановка задачі створення тренажеру для дистанційного курсу з теми «оператори керування».....	37
Bah Abibu, Parfonova T. O. Algorithm of the simulator on the topic “a straight line in space”.....	40
Рудяга Д.С., Ємець О.О. Програмне забезпечення в методі аналізу ієрархій для системного аналізу при виборі альтернативних рішень в «ПП Ткаченко А.С.».....	45
Кильник В.В., Олексійчук Ю.Ф. Програмна реалізація елементів тренажеру з теми «навчання елементарного персептрону» дисципліни «нейронно-мережеві технології в інформатиці».....	54
Доба А.І., Ємець О.О. Програмне забезпечення, що реалізує метод бажаної точки в прийнятті рішень.....	59
Задорожній А.В., Чілікіна Т.В. Розробка елементів тренажеру за темою «Параметри лінійної регресії» з дисципліни «Обчислювальні методи».....	61
Алфавітний показчик авторів.....	64

УДК 004.588

**РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕНАЖЕРА
З ТЕМИ «МЕТОД ЛЕНДА ТА ДОЙГА» ДИСТАНЦІЙНОГО
КУРСУ «МЕТОДИ ОПТИМІЗАЦІЇ ТА ДОСЛІДЖЕННЯ
ОПЕРАЦІЙ»**

О. Ю. Сивокінь, студент гр. І-41, спеціальності

«Інформатика»

Вищий навчальний заклад Укоопспілки «Полтавський
університет економіки і торгівлі»

sivokon94@gmail.com

О. О. Ємець, д.ф.-м.н., професор

Вищий навчальний заклад Укоопспілки «Полтавський
університет економіки і торгівлі»

yemetsli@ukr.net

В доповіді розглядається запропонований алгоритм і написаний по ньому тренажер на тему «Метод Ленда та Дойга» для дистанційного курсу «Методи оптимізації та дослідження операцій».

Syvokin O. Y., Iemets O.O. The construction simulator on “The method of Land and Doig” distance learning course “Methods of optimization and operations research” is considered.

Ключові слова: ТРЕНАЖЕР, ЦІЛОЧИСЛОВЕ ПРОГРАМУВАННЯ, МЕТОД ЛЕНДА ТА ДОЙГА, ЗЛП, ЗЦЛП.
Keywords: SIMULATOR, DIGITAL PROGRAMMING, LAND AND DOIG, PLP, PILP.

В тезах викладена постановка задачі і алгоритм роботи тренажера. Метою роботи є розробка алгоритму та програмного забезпечення тренажера з теми «Метод Ленда та Дойга» дистанційного навчального курсу «Методи оптимізації та

дослідження операцій». При розробці тренажера були використані лекції з дисципліни «Методи оптимізації та дослідження операцій» [1]. В [2] розглянуто деякі тренажери для дистанційного курсу «Методи оптимізації та дослідження операцій». Як в дистанційному курсі з «Методів оптимізації та дослідження операцій» ПУЕТ, так і в Інтернеті (в україномовному сегменті) взагалі немає тренажера на тему «Метод Ленда та Дойга». Тому створення такого тренажеру є актуальним. В тренажері імітується процес розв'язування задачі цілочислового лінійного програмування на основі заданого прикладу. Розроблено алгоритм, який реалізований у вигляді програми. Алгоритм представлено далі.

Крок 1. Перед студентом відкривається вікно, на якому ставиться питання: «Для яких задач застосовується метод Ленда та Дойга», і пропонується чотири варіанта відповіді а)ЗЛП; б)ЗЦЛП, в)ЗДЛП; г)ЗНП. Якщо вибрана відповідь, студент натискає на кнопку «Відповідь». Якщо вибрана відповідь правильна «б)ЗЦЛП», то відбувається перехід до наступного кроку. Якщо відповідь не правильна, з'являється інформативне вікно про неправильність відповіді, студент його закриває і зробивши відповідні висновки, вибирає правильну відповідь.

Крок 2-й. Перед студентом відкривається наступне вікно, де наведений приклад задачі (рис. 1).

$$\begin{array}{l}
 x_1 + x_2 \rightarrow \max \\
 \text{за умов} \\
 \left\{ \begin{array}{l}
 x_1 + x_3 = 5; \\
 x_1 + 6x_2 + x_4 = 12; \\
 x_1 \geq 0; \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0; \\
 x_1, x_2, x_3, x_4 - \text{цілі.}
 \end{array} \right.
 \end{array}$$

Рисунок 1. Задача на крокові 2.

На екрані з'являється питання, чи можна задачу розв'язати методом Ленда та Дойга.

Після відповіді відбувається перевірка і, перехід до наступного кроку. Якщо відповідь вибрана не вірна, тобто, «Ні», з'являється інформативне вікно про невірність відповіді, яке

потрібно закрити і вибрати коректний варіант. Якщо вибрана відповідь «Так», то відбувається перехід до наступного кроку.

Крок 3-й. Студенту задається питання, яке відноситься до задачі, яка наводилася в «Кроці 2», «Який перший крок у розв'язування даної задачі», і наведені 3 варіанта відповіді: 1) Створити ще одну нерівність; 2) Ввести нову змінну з від'ємним знаком; 3) Створити допоміжну ЗЛП відкинувши умову цілочисловості. Після вибору 3-го правильного варіанту, студент переходить до наступного кроку тренажера, якщо вибір не вірний, студент залишається на третьому кроці до тих пір, поки не вибере правильний варіант.

Крок 4-й. На даному етапі студенту дається можливість ввести дані в симплекс-таблицю згідно поставленого прикладу, який видно на екрані. Після заповнення таблиці, студент натискає кнопку «Перевірка», щоб перевірити заповнення таблиці на правильність. Якщо введені дані вірні, то студенту висвітлюється інформативне вікно, про те що данні введено правильно, і потрібно відповісти на поставлене запитання, яке стосується розв'язку таблиці, що знаходиться нижче, а саме «В таблиці міститься оптимальний розв'язок». Якщо відповідь вибрана правильна («Ні»), то переходимо до наступного кроку. Якщо ж не правильна, з'являється інформаційне вікно про не вірність і студент змушений робити висновки і давати правильну відповідь.

Крок 5-6-й. На цих двох кроках студент продовжує вводити дані в нові симплекс таблиці спираючись на результати симплекс таблиці розв'язаної на крокові 4. Після перевірки коректності введених даних в таблицю, студенту також потрібно буде відповісти на поставлені запитання, що знаходяться нижче таблиці, як і в крокові 4. Коли студент відповість на поставлені запитання, (для 5-го кроку «Який стовбець напрямний»; для 6-го кроку «Виконався критерій оптимальності»), він матиме змогу перейти до наступного кроку.

Крок 7-й. В новому вікні треба ввести розв'язок ЗЛП, а саме: « $x_1, x_2, x_3, x_4, F_{\max}$ », яку студент розв'язував протягом 4-6 кроків. Після введення даних, студент натискає кнопку «Відповідь», і в разі правильності введених даних, відбувається

перехід до наступного кроку. У разі не вірності введених даних, висвітлюється вікно помилки і студент змушений вписувати правильні.

Крок 8-й. На цьому етапі треба розбити задачу на дві по заданому критерію. Після введених даних студентом, у разі їх правильності, переходимо до наступного кроку. У разі не вірності, залишаємось на цьому кроці.

Крок 9-й. Після того, як задачу розгалузили на дві. Студенту ставиться завдання розв'язати ЗЦЛП1, тим самим методом, що і в попередніх кроках. Після того, як дані введені і зроблена перевірка, студент відповідає на запитання, яка знаходить нижче, а саме «В таблиці міститься оптимальний розв'язок?». Після обраної правильної відповіді («Ні») переходимо до наступного кроку.

Крок 10-11-й. На цих двох кроках студент продовжує вводити дані в нові симплекс-таблиці, спираючись на дані симплекс-таблиці, розглянутої на 9-му кроці. Після перевірки коректності введених даних в таблицю, студенту також потрібно буде відповісти на поставлені запитання, що знаходяться нижче таблиці, як на кроці 9 (для 10-го кроку «Виберіть напрямний стовбець»; для 11-го кроку «В таблиці міститься оптимальний розв'язок»). Коли студент відповість на поставлені запитання, він матиме змогу перейти до наступного кроку.

Крок 12-й. З'являється вікно, де треба ввести « x_1 , x_2 , x_3 , x_4 , F_{\max} » для ЗЛП, яку студент розв'язував на 9-11 кроці. Після введення даних, студент натискає кнопку «Відповідь» і у разі правильності введених даних, відбувається перехід до наступного кроку. У разі не вірності введених даних, висвітлюється вікно помилки і студент змушений вписувати правильні.

Крок 13-й. Починаємо розв'язувати ЗЦЛП2. Після того як дані введені в таблицю і зроблена перевірка, студент відповідає на запитання, яке знаходить нижче, а саме «Який стовбець напрямний?». Після обраної правильної відповіді переходимо до наступного кроку. Якщо відповідь вибрана невірна, студент залишається на 13-му кроці.

Крок 14-15-й. На цих двох кроках студент продовжує вводити дані в нові симплекс-таблиці спираючись на результати симплекс-таблиці розв'язаної на 13-му кроці. Після перевірки коректності введених даних в таблицю, студенту також потрібно буде відповісти на поставлені запитання, що знаходяться нижче таблиці, як і в крокові 13. Коли студент правильно відповість на поставлені запитання (для 14-го кроку «Який рядок напрямний?»; для 15-го кроку «Чи виконався критерій оптимальності?»), він матиме змогу перейти до наступного кроку.

Крок 16-й. В даному вікні треба ввести « x_1 , x_2 , x_3 , x_4 , F_{2max} » для ЗЛП, яку студент розв'язував протягом 13-15 кроків, а також порівняти, яка з « F_{max} » більша. Після введення даних, студент натискає кнопку «Відповідь», і у разі правильності введених даних, робота програми завершена. У разі не вірності введених даних, висвітлюється вікно помилки і студент змушений вписувати правильні.

В доповіді розглянуто реалізацію тренажера для вивчення методу Ленда та Дойга в курсі «Методи оптимізації та дослідження операцій».

Список використаних джерел

1. Ємець О. О. Методи оптимізації та дослідження операцій [Електронний ресурс] / О. О. Ємець, Т. О. Парфьонова. – Полтава : ПУЕТ, 2013. – Режим доступу: http://elib.puet.edu.ua/action.php?kt_path_info=lm.web.view&fDocumentId=670571.
2. Ємець О. О. Інформатика та системні науки (ICN-2015) [Електронний ресурс] : матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.) / О. О. Ємець. – Полтава : ПУЕТ, 2015. – Режим доступу: <http://dspace.puet.edu.ua/bitstream/123456789/2488/1>.

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
ТРЕНАЖЕРА З ТЕМИ «НЕОДНОРІДНІ
ДИФЕРЕНЦІАЛЬНІ РІВНЯННЯ ДРУГОГО ПОРЯДКУ ІЗ
СТАЛИМИ КОЕФІЦІЄНТАМИ» З ДИСЦИПЛІНИ
«МАТЕМАТИЧНИЙ АНАЛІЗ»**

*Е. Г. Безмєнов, студент КН м-51, ПУЕТ
me@operator555.su*

*В публікації розглядається розробка тренажеру за темою
«Неоднорідні диференціальні рівняння другого порядку із
сталими коефіцієнтами».*

*Е. G. Bezmiєnov. Development of the of simulators for theme
"Inhomogeneous differential equations of the second order with
constant coefficients". In the article the creating elements simulator.*

*Ключові слова: ДИФЕРЕНЦІАЛЬНІ РІВНЯННЯ, ТРЕНАЖЕР
Keywords: DIFFERENTIAL EQUATION, SIMULATOR*

Основною метою роботи є створення елементів тренажеру за темою: «Неоднорідні диференціальні рівняння другого порядку із сталими коефіцієнтами». Тренажер призначений для закріплення матеріалу і відпрацювання технічних навичок розв'язання задач.

Дана тема є однією із складних тем з дисципліни «Математичний аналіз». Тренажер реалізовується з використанням мови програмування Java, це дає можливість його запуску на таких операційних системах як: Windows, Mac OS, Solaris і дистрибутивах Linux, а також підключенню до дистанційного курсу з вказаної дисципліни.

Неоднорідні диференціальні рівняння – важливий розділ в математичного аналізу, який розглядає способи розв'язування та теорію диференціальних рівнянь. Вони широко використовуються при моделюванні різноманітних задач економічного та технічного напрямку.

Тренажер передбачатиме опитування користувача. Студент

отримуватиме завдання і покроково їх розв'язуватиме. Допуском до наступного етапу є правильна відповідь на попередній. Деякі варіанти відповідей обираються зі списку, деякі вводяться з клавіатури. На кожну тему встановлюється допустимий ліміт помилок. В разі його вичерпання відкривається довідковий матеріал у вигляді лекції згідно теми. Після її опрацювання студент повертається в режим роботи з програмою і отримує нове завдання. Після вдалого завершення розв'язування всіх етапів завдання надаватиметься інформація про успішне виконання і інформація по кількості помилок. І студент зможе отримувати наступне завдання.

При генеруванні завдань головною умовою є цілі значення, що зосереджує увагу студента на запис правильної структури відповіді.

Отже в процесі створення тренажеру для закріплення матеріалу і відпрацювання технічних навичок розв'язання задач з теми «Неоднорідні диференціальні рівняння другого порядку із сталими коефіцієнтами».

Література

1. Овчинніков П.П. Вища математика: Підручник / Овчинніков П.П., Яремчук Ф.П., Михайленко В.М. – Ч.2. – К.: Техніка, 1999. – 592с.
2. Филиппов А.Ф. Сборник задач по дифференциальным уравнениям. – М.: Наука, 1985. – 128с.

**ПРОГРАМНА РЕАЛІЗАЦІЯ ЕЛЕМЕНТІВ ТРЕНАЖЕРУ З
ТЕМИ «СОРТУВАННЯ БУЛЬБАШКАМИ» ДИСЦИПЛІНИ
«АНАЛІЗ АЛГОРИТМІВ»**

Ю. Ф. Олексійчук к. ф.-м. н., доц.

В. О. Голубенко, студент КН-51

Полтавський університет економіки і торгівлі
vladyslavholubenko@gmail.com

У публікації розглядається реалізація алгоритму тренажеру з теми «Сортування бульбашками»

V. O. Golubenko Program realization of elements of the simulator on the topic "Sorting by bubbles" discipline "Analysis of algorithms". The publication deals with the implementation of the algorithm simulator on the topic "Sorting bubbles"

*Ключові слова: МАСИВ, СОРТУВАННЯ, ТРЕНАЖЕР.
Keywords: ARRAY, SORTING, SIMULATOR.*

Розробка тренажерів навчальних дисциплін у рамках дипломних та курсових проектів студентами кафедри Математичного моделювання та соціальної інформатики Полтавського університету економіки і торгівлі є досить актуальним завданням про що свідчать ряд публікацій [1-3].

У публікації розглядається алгоритм роботи тренажера з теми «Сортування бульбашками» [4] дисципліни «Аналіз алгоритмів», що розробляється в рамках дипломної роботи. При завантаженні програмного продукту, що реалізує тренажер з теми «Сортування бульбашками» дисципліни «Аналіз алгоритмів» користувачу відображується вікно з інформацією про тему, автора та теоретичний матеріал з теми. Для початку тренінгу користувачу необхідно натиснути кнопку «Почати тренінг». Після чого користувачу будуть відображатися

питання та завдання з сортування одновимірного масиву. Якщо користувач з першої спроби не відповів вірно на поставлене йому запитання йому буде запропонована наступна спроба, якщо ж і вона виявиться невдалою то користувачу відобразиться повідомлення з підказкою. Оскільки, розроблений алгоритм є досить детальним та містить більше двадцяти кроків розглянемо типові кроки алгоритму.

Крок 1. Користувачу відображується запитання «Яке із запропонованих тверджень відповідає суті алгоритму сортування «бульбашками»?», та наводяться варіанти відповіді:

- Знаходять і вибирають в масиві елементів елемент з мінімальним значенням на інтервалі від 0 (першого) до $N-1$ (останнього) елемента і міняють його місцями з першим (0) елементом. На другому кроці знаходять елемент з мінімальним значенням на інтервалі від другого (1) до останнього ($N-1$) елемента і міняють місцями його з другим (1) елементом. І так далі для всіх елементів до $N-1$.

- При перегляді вхідної множини попарно порівнюються сусідні елементи множини. Якщо порядок їхнього проходження не відповідає заданому критерію впорядкованості, то елементи міняються місцями. В результаті одного такого перегляду при сортуванні за збільшенням елементів елемент з найбільшим значенням ключа переміститься на останнє місце в множині.

- Алгоритм складається з побудови впорядкованого бінарного дерева і подальшого його обходу.

Якщо користувач обрав *другий* варіант відповіді то відбувається перехід на наступний крок, в іншому випадку відобразиться повідомлення про помилку: «Сортування «бульбашкою» передбачає перегляд вхідної множини попарно порівнюються сусідні елементи множини. Якщо порядок їхнього проходження не відповідає заданому критерію впорядкованості, то елементи міняються місцями. В результаті одного такого перегляду при сортуванні за збільшенням елементів елемент з найбільшим значенням ключа переміститься на останнє місце в множині.»

Крок 2. Користувачу відображується завдання та запитання «Нехай дано масив з 6 цілих чисел, які необхідно розташувати в порядку зростання.

4 3 5 8 6 2

Впорядковувати необхідно зліва направо. Із запропонованих варіантів дій оберіть той, що відповідає першому крокові сортування» та наводяться варіанти відповіді:

- Порівняємо 4 і 3.
- Порівняємо 4 і 2.
- Порівняємо 3 і 5.

Якщо користувач обрав *перший* варіант відповіді то відбувається перехід на наступний крок, в іншому випадку відобразиться повідомлення про помилку: «Порівняємо 4 і 3. Число 4 більше (а значить повинно стояти правіше) - міняємо місцями ці елементи.»

Крок 3. Користувачу відображується умова та запитання «У активні комірки впишіть вірні значення першого кроку сортування» користувачу доступні комірки для введення даних (рис. 1). Якщо користувач не вірно ввів дані то відображується повідомлення про помилку: «Порівняємо 4 і 3. Число 4 більше (а значить повинно стояти правіше) - міняємо місцями ці елементи.»

4	3	5	8	6	2
3	4	5	8	6	2

Рисунок 1. – Сортування «бульбашками»

Крок 4. Користувачу відображується завдання, результати сортування з попереднього етапу та запитання «Із запропонованих алгоритмів оберіть той, що відповідає наступному крокові сортування» та наводяться варіанти відповіді:

- Порівняємо 5 і 8.
- Порівняємо 4 і 5.
- Порівняємо 6 і 2.

Якщо користувач обрав *другий* варіант відповіді то відбувається перехід на наступний крок, в іншому випадку

відобразиться повідомлення про помилку: «Числа в парі стоять в «правильному» порядку - меншу лівіше більшого. Нічого не робимо з цими елементами.»

Крок 5. Користувачу відображується умова та запитання «У активні комірки впишіть вірні значення поточного кроку сортування» користувачу доступні комірки для введення даних (рис. 2). Якщо користувач не вірно ввів дані то відображується повідомлення про помилку: «Порівняємо 4 і 5. Числа в парі стоять в «правильному» порядку - меншу лівіше більшого. Нічого не робимо з цими елементами. Масив залишається: 4 3 5 8 6 2»

4	3	5	8	6	2
3	4	5	8	6	2
3	4	5	8	6	2

Рисунок 2. – Сортування «бульбашками»

Наступні кроку алгоритму є типовими: користувач аналізує пари та сортує їх. При завершенні сортування користувач відповідає на теоретичні питання.

Крок 19. Користувачу відображується завдання, результати сортування та запитання «Скільки необхідно зробити застосувань алгоритму «бульбашки», щоб відсортувати заданий масив із N елементів?» та наводяться варіанти відповіді:

- Досить зробити N проходів, що дорівнює кількості елементів в масиві.

- Досить зробити $N-2$ прохід, оскільки коли всі числа крім останніх двох будуть на своїх місцях, остання ж пара само собою буде відсортованою.

- Досить зробити $N-1$ прохід, оскільки коли всі числа крім одного будуть на своїх місцях, останнє теж само собою опиниться на своєму місці.

Якщо користувач обрав *третій* варіант відповіді то відбувається перехід на наступний крок, в іншому випадку відобразиться повідомлення про помилку: «Досить зробити $N-1$

прохід, оскільки коли всі числа крім одного будуть на своїх місцях, останнє теж само собою опиниться на своєму місці.»

Крок 20. Користувачу відображується завдання, результати сортування та запитання «Скільки необхідно зробити порівнянь у кожному проході алгоритму «бульбашки», щоб відсортувати заданий масив із N елементів?» та наводяться варіанти відповіді:

- N порівняння на кожному проході
- $N-1$ порівняння на кожному проході.
- $N-2$ порівняння на кожному проході.

Якщо користувач обрав *другий* варіант відповіді то відбувається перехід на наступний крок, в іншому випадку відобразиться повідомлення про помилку: «Для успішного сортування треба зробити $N-1$ прохід, по $N-1$ порівняння на кожному проході.»

Крок 21. Користувачу відображується завдання, результати сортування та запитання «Для успішного сортування методом «бульбашки» треба зробити $N-1$ прохід, по $N-1$ порівняння на кожному проході. Визначте кількість операцій та складність роботи алгоритму.» та наводяться варіанти відповіді:

- Загальна кількість операцій становить $(N-1)+(N-1)$, складність алгоритму – $O(n)$.
- Загальна кількість операцій становить $(N-1) \cdot (N-1)$, складність алгоритму – $O(n^2)$.
- Загальна кількість операцій становить $(N-1)^{(N-1)}$, складність алгоритму – $O(n^2)$.

Якщо користувач обрав *другий* варіант відповіді то відбувається перехід на наступний крок де користувачу повідомляється про завершення тренінгу та кількість допущених помилок при його проходженні. Користувач може повторити тренінг або завершити роботу програми. Якщо користувач допустив помилку, то відобразиться повідомлення: «Для успішного сортування треба зробити $N-1$ прохід, по $N-1$ порівняння на кожному проході. Таким чином, загальна кількість операцій становить $(N-1) \cdot (N-1)$. Складність алгоритму сортування бульбашкою можна оцінити як $O(n^2)$.»

Література

- 1.Ємець О.О. Про розробку тренажерів для дистанційних курсів кафедрою ММСІ ПУЕТ / О.О. Ємець // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.). – Полтава: ПУЕТ, 2015. – С. 152-161.
- 2.Олексійчук Ю. Ф. Розробка та впровадження дистанційного курсу з дисципліни «Програмування» / Ю. Ф. Олексійчук // Дистанційна освіта: забезпечення доступності та неперервної освіти впродовж життя (e-learning and university education-2017): матеріали XLII Міжнародної науково-методичної конференції (м. Полтава, 9–10 лютого 2017 року) – Полтава: ПУЕТ, 2017. – С. 167-169.
- 3.Ольховська О. В. Технології підтримки системи дистанційного навчання в Полтавському університеті економіки і торгівлі / О. В. Ольховська, Д. М. Ольховський // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – С. 219-221.
- 4.Бивойно П.Г. Робота з масивами та структурами на С, С++. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Програмування» для студентів напрямів підготовки 6.050102 – „Комп’ютерна інженерія”, 6.050103 – “Програмна інженерія”, частина перша. / П.Г. Бивойно, Т.П. Бивойно, Н.О. Гора – Чернігів: ЧДТУ, 2013. – 78 с.

УДК 519.8

АЛГОРИТМІЗАЦІЯ МЕТОДУ ЗАДОВОЛЕННЯ ВИМОГ ПРИ ПРИЙНЯТІ РІШЕНЬ ЗА УМОВ ВИЗНАЧЕНОСТІ

В.О. Дяченко, студент групи І-41

Вищий навчальний заклад Укоопспілки

«Полтавський університет економіки і торгівлі»

Dyashe9812@gmail.com

У доповіді розглядається алгоритм розв'язування задачі багатокритеріальної оптимізації методу задоволення вимог при прийнятті рішень за умов визначеності.

Diachenko V.O. The paper considers the algorithm of solving the problem of multicriteria optimization of the method of satisfaction of decision-making in the conditions of certainty.

Ключові слова: ЗАДАЧА БАГАТОКРИТЕРІАЛЬНОЇ ОПТИМІЗАЦІЇ, ПРИЙНЯТТЯ РІШЕНЬ, МЕТОД ЗАДОВОЛЕННЯ ВИМОГ, ГОЛОВНИЙ КРИТЕРІЙ.

Keywords: THE PROBLEM OF MULTI-CRITERIAN OPTIMIZATION, DECISION-MAKING, METHOD OF REQUIREMENT SATISFACTION, MAIN CRITERIA.

В тезах викладений алгоритм вирішення багатокритеріальної задачі методом задоволення вимог при прийнятті рішень за умов визначеності.

Вступ: Прийняття рішення в багатокритеріальних задачах полягає у свідомому виборі з множини альтернатив однієї. Цей вибір робить особа, що приймає рішення (ОПР), яка прагне до досягнення своєї певної цілі. У ролі такої особи виступають чи окрема людина (прийняття індивідуальних рішень), чи група людей (прийняття колективних рішень), що володіють правами вибору рішення і несуть відповідальність за його наслідки.

Для задач прийняття рішень в умовах визначеності, коли випадкові та невизначені фактори відсутні, компонентами такої моделі є: множина X альтернатив (рішень), із якої і варто зробити вибір однієї найкращої альтернативи (оптимального розв'язку), і опис переваг особи, що приймає рішення. Для того, щоб була забезпечена можливість (воля) вибору, множина альтернатив X повинна містити не менш двох елементів [1].

Розв'язування задачі відбувається в декілька кроків.

ОПР виділяє "головний критерій" $f_i(x)$, $i \in \{1, \dots, m\}$, який, на його думку, найбільш за всі інші повинен бути покращеним. Далі встановлює мінімально допустимі рівні значень інших критеріїв $\xi_j^k, j = \overline{1, m}, j \neq i$.

У наслідок розв'язку однокритеріальної задачі:

$$\max_{x \in G_k} f_i(x), \text{ де } G_k = \{x \in X \mid f_j(x) \geq \xi_j^k, j = \overline{1, m}, j \neq i\}, \quad (1)$$

визначається ефективна альтернатива x^k і її оцінка.

$$y^k = (f_1(x^k), \dots, f_m(x^k)) \quad (2)$$

ОПР аналізує отримане значення головного критерію. Якщо воно не задовольняє його, то переходить на наступний крок, залишаючи номер головного критерію незмінним. Якщо значення головного критерію задовольняє ОПР, то він розмірковує – можливо чи ні деяке погіршення значення головного критерію з метою покращення значень інших. Якщо – "ні", то процедура закінчується. У протилежному випадку – переходить на наступний крок із метою призначити інший головний критерій.

Метод використовує два типи інформації від ОПР: інформацію про домінування одного критерію над іншими й інформацію про діапазони значень критеріїв [1].

Отриману однокритеріальну задачу вирішуємо одним із методів розв'язку задач лінійного програмування: симплекс-методу або методу штучного базису.

Щоб розв'язати задачу лінійного програмування за допомогою симплекс-методу, необхідно виконати такі етапи :

1) визначити початковий опорний план задачі лінійного програмування;

2) побудувати симплекс таблицю;

3) перевіряють опорний план на оптимальність за допомогою оцінок. Якщо план не є оптимальним, то переходять до нового опорного плану або встановлюють, що оптимального плану не існує;

4) переходять до нового опорного плану задачі, тобто розраховують нову симплексну таблицю;

5) повторюють дії, починаючи з п. 3.

На етапі визначення початковою опорного плану задачі лінійного програмування, після її зведення до канонічної форми, відшукують m одиничних лінійно незалежних векторів, які становлять базис m - вимірного простору. Можливі такі випадки:

- є необхідна кількість одиничних векторів, тоді початковий опорний план визначається безпосередньо без додаткових дій;

- у системі обмежень немає необхідної кількості одиничних незалежних векторів. Тоді для побудови першого опорного плану застосовують метод штучного базису [2].

Висновок: Процес вирішення задачі багатокритеріальної оптимізації займає досить багато часу, а таких задач в різних сферах діяльності дуже велика кількість. Тому вирішення таких задач не може обійтися без засобів автоматизації. Вирішення такої задачі за допомогою програмного забезпечення, являє собою ітеративний процес взаємодії між ОПР і програмним забезпеченням.

За описаним алгоритмом розроблено програмне забезпечення. Проведено тестування, результати якого, підтверджують працездатність розробленого програмного забезпечення і дозволяють рекомендувати його для використання на практиці при вирішенні задач багатокритеріальної оптимізації при прийнятті рішень за умов визначеності.

Література

1. Волошин, О. Ф. Моделі та методи прийняття рішень : навч. посіб. для студ. вищ. навч. закл. / О. Ф. Волошин, С.О. Машенко. – 2-ге вид., перероб. та допов. – К. : Видавничо-поліграфічний центр "Київський університет", 2010. – 336 с.
2. Ємець О.О. Навчально методичний посібник з курсом лекцій "Методи оптимізації та дослідження операцій" / О.О. Ємець, Т.О. Парфьонова. – Полтава: РВВ ПУЕТ, 2013. – 492 с.
3. Рогоза М.Є. Системи підтримки прийняття рішень: навч. пос. / М.Є. Рогоза, О.О. Ємець, Є.М. Ємець. – Полтава: РВВ ПУЕТ, 2013. – 328 с.

УДК 004.588

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ ЕЛЕМЕНТІВ
ТРЕНАЖЕРА З ПОБУДОВИ МАТЕМАТИЧНОЇ МОДЕЛІ
КОМБІНАТОРНОЇ ОПТИМІЗАЦІЙНОЇ
“ЗАДАЧІ ДИРЕКТОРА”**

В. В. Куркін, студент гр. І-41, спеціальності «Інформатика»
Вищий навчальний заклад Укоопспілки «Полтавський
університет економіки і торгівлі»
caucho228@gmail.com

О. О. Ємець, д.ф.-м.н., професор
Вищий навчальний заклад Укоопспілки «Полтавський
університет економіки і торгівлі»
yemetsli@ukr.net

*Запропоновано алгоритм і створено тренажер на
тему «Задача директора» для дистанційного курсу «Методи
оптимізації та дослідження операцій».*

*Kyrkin V. V., Iemets O.O. ALGORITHMIZATION AND
PROGRAMMING OF ELEMENTS OF TRAINER OF
CONSTRUCTION OF MATHEMATICAL MODEL OF
COMBINATOR OPTIMIZATION “DIRECTOR’S PROBLEM”.*

*An algorithm is proposed and a trainer is created on the topic
"Director's problem" for the distance course "Methods of
optimization and operations research ".*

Ключові слова: АЛГОРИТМ, ТРЕНАЖЕР, ЗАДАЧА
ДИРЕКТОРА, ПРОГРАМУВАННЯ.

Keywords: ALGORITM, TRAINER, PROGRAMMING,
DIRECTOR'S PROBLEM.

В доповіді викладена постановка задачі і частина
алгоритму роботи тренажера. Метою роботи є розробка
алгоритму та програмного забезпечення тренажера з теми

«Задача директора» дистанційного навчального курсу «Методи оптимізації та дослідження операцій». Тренажер створюється для вивчення цієї теми. При ознайомленні з темою «Задача директора» були використані лекції з дисципліни «Методи оптимізації та дослідження операцій»[1]. Як в дистанційному курсі з «Методів оптимізації та дослідження операцій» ПУЕТ, так і в Інтернеті (в україномовному сегменті) немає тренажера на тему «Задача Директора». Тому створення такого тренажера є актуальним. В тренажері повинно імітуватися процес розв'язування задачі. Алгоритм представлено далі.

Алгоритм тренажера.

Крок 1. На екрані умова задачі (у подальшому умова має буди завжди на екрані до завершення складання моделі). В приймальні k осіб, час прийому яких відомий: T_1, \dots, T_k відповідно. Директору треба так організувати прийом, щоб загальний час очікування був мінімальний.

Далі – запитання:

Що необхідно визначити в результаті задачі?

а) X_j час, що йде на прийом відвідувача, що зайшов останнім;

б) X_j час, що йде на прийом відвідувача, що зайшов першим;

в) X_j час, що йде на прийом відвідувача, що зайшов j -им;

г) час, що йде на прийом k осіб;

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь. (X_j час, що йде на прийом відвідувача, що зайшов j -им;)

Інакше перехід до наступного кроку.

Крок 2-й. З'являється наступне питання: “Якщо зайшов перший відвідувач, скільки залишилося в приймальні?”

а) 1;

б) 2;

в) $k-1$;

г) $k+1$.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: $k-1$.

Інакше перехід до наступного кроку.

Крок 3-й. З'являється наступне питання: “ Скільки часу чекає кожен відвідувач, коли зайшов перший?”

а) T_1 ;

б) T_j ;

в) X_j ;

г) X_1 .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: X_1 .

Інакше перехід до наступного кроку.

Крок 4-й. З'являється наступне питання: “ Скільки часу чекають усі відвідувачі, коли зайшов перший відвідувач?”

а) $(k-1)*T_1$;

б) $(k-1)*X_1$;

в) X_1 ;

г) T_1 .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: $(k-1)*X_1$.

Інакше перехід до наступного кроку.

Крок 5.

З'являється наступне питання: “ Скільки часу чекає кожен відвідувач, коли приймають j -го відвідувача?”

а) X_j ;

б) X_1 ;

в) T_1 ;

г) T_j .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: X_j .

Інакше перехід до наступного кроку.

Крок 6.

З'являється наступне питання: “ Скільки часу чекають усі відвідувачі, коли зайшов j-ий відвідувач?”

а) $(k-1)*X_1$;

б) $(k-j)*X_j$;

в) $(k-j)*X_1$;

г) $(k-j)*T_1$.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: $(k-j)*X_j$

Інакше перехід до наступного кроку.

Крок 7.

З'являється наступне завдання: “Який вигляд має цільова функція? (Внесіть в активні комірки значення цільової функції)”

Потрібно в порожні комірки вписати певне значення.

$(\text{---}) * X_1 + (\text{---}) * X_2 + \dots + (\text{---}) * X_j + \dots + (\text{---}) * X_{k-1} + (\text{---}) * X_k$

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: $(k-1)*X_1 + (k-2)*X_2 + \dots + (k-j)*X_j + \dots + (1)*X_{k-1} + (0)*X_k$.

Інакше перехід до наступного кроку.

Крок 8.

З'являється наступне питання: “Цільова функція буде мінімізуватися чи максимізуватися?”

а) Мінімізуватися;

б) Максимізуватися.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: мінімізуватися.

Інакше – перехід до наступного кроку.

Крок 9.

З'являється таке питання: “ Яка з осіб може прийматися першою?”

а) Та у якої час обслуговування T_1 ;

б) Будь-хто;

в) Та у якої час прийому найменший.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: будь-хто.

Інакше перехід до наступного кроку.

Крок 10.

З'являється наступне питання: “Яка з осіб може прийматися наступною?”

а)Будь-хто;

б)Будь-хто з тих, що залишилися;

в)Та особа у якої час прийому T_2 .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: будь-хто з тих, що залишилися.

Інакше перехід до наступного кроку.

Крок 11.

З'являється наступне питання: “Яка з осіб може прийматися j-ою?”

а)Та особа, у якої час прийому T_j ;

б)Будь-хто;

в)Будь-хто з тих, що залишилися.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: будь-хто з тих, що залишилися.

Інакше перехід до наступного кроку.

Крок 12.

З'являється наступне питання: “Отже, з якої комбінаторної множини вибирається вектор невідомих $x=(X_1,...,X_k)$?”.

а)Множина перестановок;

б)Множина розміщень, де $T=\{T_1,...,T_k\}$, а v -кількість різних елементів в T .

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: множина перестановок.

Інакше перехід до наступного кроку.

Крок 13.

З'являється наступне питання: “Чи є інші обмеження в задачі директора?”.

а) Ні;

б) Так.

Користувач має обрати одну правильну відповідь. Якщо вибір не правильний, то з'явиться повідомлення про помилку, яке містить повну відповідь: ні.

Інакше з'являється вікно з повідомленням про завершення роботи тренажера.

Розроблений алгоритм реалізовано у вигляді програми на мові C# у програмному середовищі Visual Studio 2017 Community, яка містить близько 1000 рядків програмного коду.

Розроблено новий тренажер по створенню математичної моделі «Задачі директора» для дистанційного курсу «Методи операцій та дослідження операцій». Цей тренажер можна використовувати як в дистанційному так і в стаціонарному навчанні студента.

Список використаних джерел

1. Ємець О.О. Методи оптимізації та дослідження операцій [Електронний ресурс] : навчально-методичний посібник за кредитно-модульною організацією навчального процесу/ О. О. Ємець, Т. О. Парфьонова. – Полтава : ПУЕТ, 2013. Режим доступу:
http://elib.puet.edu.ua/action.php?kt_path_info=lm.web.view&fDocumentId=670571
2. Ємець О.О. Елементи комбінаторної оптимізації: Навчально-методичний посібник / О.О. Ємець. – Полтава: РВВ ПУСКУ, 2009. – 23 с.
3. Рихтер Дж. Язык программирования C#. Классика Computers Science/ Дж. Рихтер. – Леверкузен: Addison Wesley, 2011. – 715 с.
4. Хейлсберг А. Изучаем C#/ А. Хейлсберг. – Нью-Йорк: Питер, 2013. – 537 с.

**ТРЕНАЖЕР З ТЕМИ «1-R АЛГОРИТМ» ДИСЦИПЛІНИ
«КОМП'ЮТЕРНИЙ АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ»**

В. М. Мандрика, студент групи КН-51

Полтавський університет економіки і торгівлі
vitalstar@bk.ru

Ю. Ф. Олексійчук, к. ф.-м. н.

Полтавський університет економіки і торгівлі
olexijchuk@gmail.com

В доповіді розглядається алгоритм роботи тренажеру з теми «1-R алгоритм» дисципліни «Комп'ютерний аналіз статистичних даних».

Mandryka V. M., Oleksiichuk Yu. F. Article An algorithm for the simulator for the subject «1-R algorithm» of the discipline «Computer analysis of statistical data» is considered.

Ключові слова: ТРЕНАЖЕР, 1-R АЛГОРИТМ, ДИСТАНЦІЙНЕ НАВЧАННЯ.

Keywords: SIMULATORS, 1-R ALGORITHM, DISTANCE LEARNING.

В Полтавському університеті економіки і торгівлі впроваджується дистанційне навчання.[1,2] Для практичної підготовки студентів важливу роль відіграють навчальні тренажери. За останні роки розроблено багато тренажерів з різних навчальних предметів [3-5]. Але розробка тренажеру з теми «1-R алгоритм» дисципліни «Комп'ютерний аналіз статистичних даних» залишається актуальною.

Розглянемо основну ідею 1-R алгоритму [6]. Цей метод для класифікації використовує лише одну незалежну змінну, тому його називають "1-правило" (1-rule) або 1-R алгоритм. Для всіх можливих значень кожної із незалежних змінних формулюється

правило, яке класифікує об'єкти із навчальної вибірки. При цьому в висновку правила залежній змінній надається значення, яке зустрічається для даної умови найчастіше. Таким чином, для кожної змінної буде отриманий набір правил для кожного значення. Далі вибирається та змінна, для якої побудовані правила з найменшою похибкою.

Незважаючи на свою простоту, інколи 1-R алгоритм може використовуватися у практичних задачах.

Тренажер призначений для ознайомлення студентів з роботою 1-R алгоритму на практиці. Перед використанням тренажеру передбачається, що студент ознайомився з теоретичним матеріалом, зокрема відповідною лекцією.

Загальний алгоритм роботи тренажеру.

Крок 0. Виводиться завдання, яке полягає у застосуванні 1-R алгоритму до деякої вибірки, яка містить одну залежні змінну і кілька незалежних. Кожна незалежна змінна має кілька градацій, залежна змінна має 2 градації (так/ні). Важливо, щоб кількість градацій для незалежних змінних була приблизно однаковою.

Крок 1. Беремо наступну незалежну змінну, якщо це можливо. Якщо всі змінні перебрані – перехід на крок 8.

Крок 2. Для кожної градації вибраної незалежної змінної студент має порахувати кількість значень «так» та кількість значень «ні» залежної змінної. Якщо відповідь правильна – перехід на крок 4, якщо ні – перехід на крок 3.

Крок 3. Студенту виводиться інформація про помилку та підказка. Перехід на крок 2.

Крок 4. Студент має вибрати правило класифікації. Якщо відповідь правильна – перехід на крок 6, якщо ні – перехід на крок 5.

Крок 5. Студенту виводиться інформація про помилку та підказка. Перехід на крок 4.

Крок 6. Студент має порахувати помилку. Помилка – це кількість неправильно класифікованих вимірів при користуванні сформульованим правилом. Якщо відповідь правильна – перехід на крок 1, якщо ні – перехід на крок 7.

Крок 7. Студенту виводиться інформація про помилку та підказка. Перехід на крок 6.

Крок 8. Студент має визначити, яка із незалежних змінних дає найкращий результат. Найкращий результат – це мінімальна кількість неправильно класифікованих вимірів. Якщо відповідь правильна – завершення алгоритму, якщо ні – перехід на крок 9.

Крок 9. Студенту виводиться інформація про помилку та підказка. Перехід на крок 8.

Для розробки тренажеру вибрана мова програмування Java, оскільки це дозволить використовувати тренажер незалежно від апаратної та операційної платформи.

В доповіді розглянуто алгоритм роботи тренажеру з теми «1-R алгоритм» дисципліни «Комп'ютерний аналіз статистичних даних».

Література

1. Ольховська О. В. Технології підтримки системи дистанційного навчання в Полтавському університеті економіки і торгівлі / О. В. Ольховська, Д. М. Ольховський // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – С. 219-221.
2. Олексійчук Ю. Ф. Розробка та впровадження дистанційного курсу з дисципліни «Програмування» / Ю. Ф. Олексійчук // Дистанційна освіта: забезпечення доступності та неперервної освіти впродовж життя (e-learning and university education-2017): матеріали XLII Міжнародної науково-методичної конференції (м. Полтава, 9–10 лютого 2017 року) – Полтава: ПУЕТ, 2017. – С. 167-169.
3. Ємець О.О. Про розробку тренажерів для дистанційних курсів кафедрою ММСІ ПУЕТ / О.О. Ємець // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.). – Полтава: ПУЕТ, 2015. – С. 152-161.
4. Чілікіна Т. В. Огляд тренажерів з дисципліни "Математичний аналіз" на прикладі розробок студентів напряму

- "Інформатика" / Т. В. Чілікіна // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – С. 329-330.
5. Русін В. С. Програмна реалізація елементів тренажеру з теми "Аналіз алгоритму сортування вставками" дисципліни "Аналіз алгоритмів" / В. С. Русін, Ю. Ф. Олексійчук // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю (м. Полтава, 16–18 березня 2017 р.) – Полтава: ПУЕТ, 2017. – С. 236-237.
6. Барсеян А. А. Методы и модели анализа данных: OLAP и Data Mining. / А. А. Барсеян, М. С. Куприянов, В. В. Степаненко, И. И. Холод – СПб.: БХВ-Петербург, 2004. – 336 с.

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ ЕЛЕМЕНТІВ
ТРЕНАЖЕРА З ПОБУДОВИ МАТЕМАТИЧНОЇ МОДЕЛІ
КОМБІНАТОРНОЇ ОПТИМІЗАЦІЙНОЇ
ЗАДАЧІ ПРО ОПТИМІЗАЦІЮ СУМІШІ**

*Д. О. Стовбун, студент гр. І-41, спеціальності «Інформатика»
Вищий навчальний заклад Укоопспілки «Полтавський
університет економіки і торгівлі»
dmatrijstovbun@gmail.com*

*О. О. Ємець, д.ф.-м.н., професор
Вищий навчальний заклад Укоопспілки «Полтавський
університет економіки і торгівлі»
yemetsli@ukr.net*

*Запропоновано алгоритм і створено тренажер на
тему «Задача про оптимізацію суміші» для дистанційного
курсу «Методи оптимізації та дослідження операцій».*

*Stovbun D. O., Iemets O.O. Algorithmization and programming of
elements of the simulator for constructing of the mathematical model
of the optimization problem of the mixture. For the topic "Task of the
mixture optimization" for the distance course "Methods of
optimization and operations research" the algorithm is proposed and
the trainer is created.*

Ключові слова: АЛГОРИТМІЗАЦІЯ, ТРЕНАЖЕР,
ОПТИМІЗАЦІЯ, МАТЕМАТИЧНА МОДЕЛЬ, ЗАДАЧА ПРО
ОПТИМІЗАЦІЮ СУМІШІ, ПРОГРАМУВАННЯ.

Keywords: ALGORITHMIZATION, SIMULATOR,
OPTIMIZATION, MATHEMATICAL MODEL, TASK FOR
OPTIMIZATION OF A MIXTURE, PROGRAMMING.

В доповіді викладена частина алгоритму роботи
тренажера та постановка задачі по його створенню. Метою

роботи є розробка алгоритму та тренажера з теми «Задача про оптимізацію суміші» дистанційного навчального курсу «Методи оптимізації та дослідження операцій». Тренажер створюється для вивчення студентами цієї теми. При розробці тренажера «Задача про оптимізацію суміші» були використані лекції з дисципліни «Методи оптимізації та дослідження операцій»[1]. В дистанційному курсі ПУЕТ немає, а у всесвітній мережі Інтернет також не знайдено тренажера на тему «Задача про оптимізацію суміші». Тому тренажер є актуальним. В тренажері повинно імітуватися процес розв'язування задачі. Алгоритм представлено далі.

Алгоритм тренажера.

Крок 1.

Відкривається вікно з назвою тренажера:

Завдання:

Скласти математичну модель задачі про оптимізацію суміші

Крок 2.

На екрані з'являється умова задачі яка залишається на екрані до кінця виконання всіх тестів.

Задача

Нехай суміш потрібно скласти з n різних видів сировини, кожний з яких містить m видів елементів(речовин), що нас цікавлять.

Нехай a_{ij} , ($i \in J_m$) – кількість i – тої речовини в одиницях j – тої сировини, вартість якої c_j , $j \in J_n$. Позначимо через b_i – найменшу допустиму кількість i – тої речовини, через d_j – запас (обсяг) сировини. Позначимо x_j – кількість сировини j – го виду, яка необхідна для складання суміші. Потрібно щоб вартість суміші була мінімальною.

Тут використано позначку $J_m = \{1, 2, \dots, m\}$.

Крок 3.

Переходимо до складання математичної моделі.

Введемо необхідні змінні. Зробіть правильний вибір:

- А) x_j – кількість j – тої речовини в суміші;
- Б) x_j – кількість j – тої сировини в суміші; +
- В) x_j – вартість одиниці j – тої сировини;
- Г) x_j – запас j – тої сировини.

Якщо відповідь правильна – перехід далі, якщо ні – підказка про неправильну відповідь, показ правильної і перехід до наступного кроку.

Крок 4.

Які складові частини має математична модель оптимізаційної задачі?

- А) тільки цільову функцію;
- Б) тільки обмеження;
- В) і цільова функція, і обмеження; +
- Г) складові на вимогу замовника моделі.

Якщо відповідь правильна – перехід далі, якщо ні – підказка про неправильну відповідь, показ правильної і перехід до наступного кроку.

Крок 5.

Складання цільової функції.

Що виражає цільова функція задачі :

- А) мінімізацію вартості суміші; +
- Б) мінімізацію вартості запасів;
- В) максимізацію кількості інгредієнтів;
- Г) максимізацію вартості суміші.

Якщо відповідь правильна – перехід далі, якщо ні підказка про неправильну відповідь, показ правильної і перехід до наступного кроку.

Крок 6

Для складання математичної моделі потрібно згадати, як позначена кількість j – тої сировини в суміші:

- А) c_j ;
- Б) d_j ;

В) x_j . +

Якщо відповідь правильна – перехід далі, якщо ні підказка про неправильну відповідь, показ правильної і перехід до наступного кроку.

Крок 7.

Для складання математичної моделі треба обрати правильну відповідь на питання:

скільки коштує одиниця j –тої сировини :

А) c_j ; +

Б) d_j ;

В) x_j .

Якщо відповідь правильна – перехід далі, якщо ні підказка про неправильну відповідь, показ правильної і перехід до наступного кроку.

Крок 8

Скільки коштує використання в суміші всієї j – тої речовини?

Відповідь (дати можливість вводу) :

$c_j * x_j$ або $x_j * c_j$ – правильні, всі інші відповіді не правильні.

Якщо відповідь правильна – поява нової комірки для вводу, якщо ні – програма сама підставить правильну відповідь і відкриє наступну комірку .

Крок 9

Як змінюється параметр j в цільовій функції ?

А) від 1 до n ; +

Б) від 1 до m ;

В) від m до n ;

Г) від 3 до 5.

Якщо відповідь правильна – перехід далі, якщо ні підказка про неправильну відповідь, показ правильної і перехід до наступного кроку.

Крок 10

Записати цільову функцію(дається можливість ввести всі елементи цільової функції та вибрати **min чи max**)

$$\sum_{j=1}^n x_j * c_j \rightarrow \min/\max(\text{вибір})$$

Якщо відповідь правильна – поява нової комірки для вводу, якщо ні – програма сама підставить правильну відповідь і відкриє наступну комірку .

Крок 11

Переходимо до складання обмеження.

Позначте обмеження, які необхідно скласти

А) По мінімальній кількості необхідної речовини; +

Б) по мінімальній вартості суміші;

В) по кількості i – тої речовини в одиниці j – тої сировини;

Г)по запасам j – тої речовини .+

Якщо відповідь правильна – перехід далі, якщо ні підказка про неправильну відповідь, показ правильної і перехід до наступного кроку.

Крок 12

Скільки i – тої речовини в усій j – тій сировині?

$$a_{ij} * x_j \quad \forall i$$

Якщо відповідь правильна – поява нової комірки для вводу, якщо ні – програма сама підставить правильну відповідь і відкриє наступну комірку .

Крок 13

Скласти обмеження

$$\sum_{j=1}^n a_{ij} x_j \geq b_i$$

Якщо відповідь правильна – поява нової комірки для вводу, якщо ні – програма сама підставить правильну відповідь і відкриє наступну комірку .

Крок 14

Скласти обмеження по запасу j – тої сировини

$$0 \leq x_j \leq d_j$$

Якщо відповідь правильна – поява нової комірки для вводу, якщо ні – програма сама підставить правильну відповідь і відкриє наступну комірку .

Був розроблений алгоритм та створений тренажер для навчання побудові математичної моделі задачі про оптимізацію суміші.

Список використаних джерел

1. Ємець О.О. Методи оптимізації та дослідження операцій [Електронний ресурс] : навчально-методичний посібник за кредитно-модульною організацією навчального процесу/ О. О. Ємець, Т. О. Парфьонова. – Полтава : ПУЕТ, 2013. Режим доступу:
http://elib.puet.edu.ua/action.php?kt_path_info=lm.web.view&fDocumentId=670571
2. Ємець О.О. Елементи комбінаторної оптимізації: Навчально-методичний посібник / О.О. Ємець. – Полтава: РВВ ПУСКУ, 2009. – 23 с.
3. Рихтер Дж. Язык программирования C#. Классика Computers Science/ Дж. Рихтер. – Леверкузен: Addison Wesley, 2011. – 715 с.
4. Хейлсберг А. Изучаем C#/ А. Хейлсберг. – Нью-Йорк: Питер, 2013. – 537 с.

ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ ТРЕНАЖЕРУ ДЛЯ ДИСТАНЦІЙНОГО КУРСУ З ТЕМИ «ОПЕРАТОРИ КЕРУВАННЯ»

О. В. Щербак, студент КН м-51

Полтавський університет економіки і торгівлі
viruscool92@gmail.com

*В доповіді розглядаються постановка задачі тренажеру
«Оператори керування»*

*Scherbak A. V. Resolving the challenge of creating trainer for a
remote course with the theme “Operators of management”. In
speech resolving the challenge of simulator “Operators of
management” was considered.*

Ключові слова: ОПЕРАТОРИ КЕРУВАННЯ.

Keywords: OPERATORS OF MANAGEMENT.

Актуальність проблеми створення тренажеру з теми «Оператори керування» на прикладі коду Java, полягає у відсутності жодного подібного тренажеру забезпеченні на сайті дистанційного курсу «Програмування І» в ПУЕТ. Через це постає необхідність у забезпеченні вищого навчального закладу відповідним програмним забезпеченням для більш якісного засвоєння знань та навичок студентами, що будуть вивчати мову Java. Необхідність створення тренажеру з теми «Оператори керування», зумовлено тим що, згідно з опитуванням 8186 людей (на 23 січня 2017 року) [4], мова Java є однією з найбільш розповсюджених мов програмування. Тому студентам необхідно надати можливість зручного способу засвоєння навчального матеріалу для даної мови програмування.

Слід зазначити, що оператори керування є основою будь якої мови програмування, оскільки, завдяки операторам

керування, відбувається процес розділення коду програмного забезпечення на блоки, для того щоб більш коректно писати код. Для більш якісного виконання поставленої задачі, було розглянуто значну кількість тренажерів та навчальних систем з програмування. Особлива увага була приділена матеріалам наведеним далі.

Перший матеріал – це JavaLearn[1]. Перевагами даного матеріалу є його безкоштовність, та можливість завантажити його з репозиторія GitHub автора. Також до переваг можна віднести те що JavaLearn допомагає користувачам поліпшити свої навички володіння мовою програмування Java. До недоліків можна віднести те, що новачкам буде важко почати працювати із JavaLearn, тому що необхідно встановлювати додаткове програмне забезпечення для роботи із даним матеріалом. Також значним недоліком є те, що JavaLearn не проводить контролю над користувачем, користувач повинен виконувати завдання до тих пір, поки він не отримає необхідну відповідь. Як висновок, JavaLearn – це швидше науковий довідник, ніж тренажер, який допоможе у вивченні мови Java.

Другий матеріал – це CodeHunt[2]. Для початку далі слід зазначити переваги цього матеріалу. CodeHunt надає можливість обрати між двома мовами програмування: C# та Java, CodeHunt контролює роботу користувача та у разі помилки у коді зверне увагу користувача. У CodeHunt зручний інтерфейс, вікно, з яким працюємо, поділено на дві частини. У правій частині відображається завдання, показано результат роботи коду та виводиться повідомлення щодо помилок. У лівій частині інтерфейсу користувач повинен вписувати рядки коду. Важливою перевагою є те, що CodeHunt не вимагає від користувача встановлення додаткового програмного забезпечення. Для роботи достатньо відкрити сторінку тренажеру в браузері. До недоліків можна віднести певну складність для початківців, оскільки тренажер вимагає певного розуміння базових операторів керування, за допомогою яких користувач повинен виконати поставлену задачу. Як висновок, слід зазначити, що тренажер є досить зручним. Він допоможе

краще оволодіти операторами керування, але у новачків можуть виникнути складності із початком роботи. Останній матеріал – це CodeCombat[3]. Для початку слід зазначити переваги даного матеріалу. Тренажер CodeCombat не вимагає знання мов програмування. Також користувачу надається можливість обрати із декількох мов програмування, таких як: JavaScript, CoffeeScript, Python, Lua. Тренажер виконано у вигляді гри, де користувач повинен провести героя від початку до кінця рівня. Якщо у користувача виникають складності, то він може звернутися до інструкції, де зможе ознайомитися з тим, що необхідно зробити, щоб пройти цей рівень та як краще побудувати код. Слід зазначити, що герой, яким керує персонаж, виконує кожен дію, що запрограмував користувач, тому, якщо щось йде не за планом, користувач побачить, в якому рядку він зробив помилку. Тренажер CodeCombat надає можливість проводити сеанси у режимі Студент-Викладач, де буде людина, що оцінює та керує діями студента. До недоліків можна віднести те, що у користувача можуть виникнути певні труднощі на більш складних рівнях, де вимагається використання циклів та виключень. Також CodeCombat пропонує використовувати вже написані методи, а не програмувати їх самостійно. Як висновок, CodeCombat є зручним тренажером для знайомства з основами програмування та підходить навіть дітям.

Література

- 1.Github [Електронний ресурс]: JavaLearn. – Режим доступу: <https://github.com/rodomod/JavaLearn>
- 2.Code Hunt [Електронний ресурс]: code hunt. – Режим доступу: <https://www.codehunt.com/>
- 3.Code Combat [Електронний ресурс]: Найзахопливіша гра для вивчення програмування. / Nick Winter / Matt Lott //. – Режим доступу: <https://codecombat.com>
- 4.DOU [Електронний ресурс]: Рейтинг зъяков програмерования №8. – Режим доступу: <https://dou.ua/lenta/articles/language-rating-jan-2017>.

UDC 004.588

**ALGORITHM OF THE SIMULATOR ON THE TOPIC
“A STRAIGHT LINE IN SPACE”**

Bah Abibu, *Student of group I-41*
Poltava University of Economics and Trade
abibubah@gmail.com

T. O. Parfonova, *Doctor of Philosophy of Physics and*
Mathematics, Assisant Professor
Poltava University of Economics and Trade

The report considers the algorithm of the simulator on the topic “A straight line in space”

Keywords: SIMULATOR, A STRAIGHT LINE,
ALGORITHM OF THE SIMULATOR

Each algorithm is analyzed with respect of computational accuracy. A comparison is given among the bounding algorithms with the aim of determining the right algorithms for correction requirements.

The report developed a simulator algorithm.

Step 1. A question appears on the screen. The user needs to choose one of the variants of answer, among which one is correct. If the right choice is made, then move to step 2. If not - an error reporting that contains the correct answer appears. Move to step 2.

How do we call the equation of a line in a space of the form

$$\begin{cases} A_1x + B_1y + C_1z + D_1 = 0, \\ A_2x + B_2y + C_2z + D_2 = 0. \end{cases} ?$$

- A) parametric;
- B) canonical;

B) general;

D) the equation of a straight line that passes through two points.

Error reporting:

An equation of a straight line of the form $\begin{cases} A_1x + B_1y + C_1z + D_1 = 0, \\ A_2x + B_2y + C_2z + D_2 = 0. \end{cases}$ – this is a general equation of a straight line, as a line of intersection of planes with normals $\vec{N}_1 = (A_1; B_1; C_1)$ и $\vec{N}_2 = (A_2; B_2; C_2)$.

Step 2. A question appears on the screen. The user needs to choose one of the answers, among which one is correct. If the right choice is made, then move to step 3. If not – an error reporting that contains the correct answer appears. Move to step 3.

How do we call the equation of a line in a space of the form

$$\frac{x - x_0}{l} = \frac{y - y_0}{m} = \frac{z - z_0}{n} \quad ?$$

A) parametric;

B) canonical;

B) general;

D) the equation of a straight line that passes through two points.

Error reporting:

An equation of a straight line of the form $\frac{x - x_0}{l} = \frac{y - y_0}{m} = \frac{z - z_0}{n}$ – is a canonical equation of the straight line, where $(x_0; y_0; z_0)$ – the point that belongs to a given line, l, m, n – the coordinates of the directing vector of the straight line.

Step 3. A question appears on the screen. The user needs to choose one of the variants of answers, among which one is correct. If

the right choice is made, then move to step 4. If not - an error reporting that contains the correct answer appears. Move to step 4.

How do we call the equation of a line in a space of the form

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} ?$$

- A) parametric;
- B) the equation of a straight line that passes through two points;
- B) general;
- D) canonical.

Error reporting:

An equation of a straight line of the form $\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1}$ – is an equation of a straight line that passes through two points $(x_1; y_1; z_1)$ и $(x_2; y_2; z_2)$

Step 4. A question appears on the screen. The user needs to choose one of the answers, among which one is correct. If the right choice is made, then move to step 5. If not - an error reporting that contains the correct answer appears. Move to step 5.

How do we call the equation of a line in a space of the

form $\begin{cases} x = x_0 + lt \\ y = y_0 + mt \\ z = z_0 + nt \end{cases} ?$

- A) canonical;
- B) the equation of a straight line that passes through two points;
- B) general;
- D) parametric.

Error reporting:

$$\begin{cases} x = x_0 + lt \\ y = y_0 + mt \\ z = z_0 + nt \end{cases} -$$

An equation of a straight line of the form $\begin{cases} x = x_0 + lt \\ y = y_0 + mt \\ z = z_0 + nt \end{cases}$ is a parametric equation of a straight line, where $(x_0; y_0; z_0)$ – is a point that belongs to a given line, l, m, n – the coordinates of the directing vector of the line, t – parameter.

Step 5. There is a task on the screen

Write the equation of a straight line that passes through two points $A(3; 2; -1)$ and $B(4; -1; 3)$

Fig. 1.

The user in one by one enters numerical values and symbols into active cells (in numerical order) (Fig. 1).

In the cell \square^2 either the value 3 or the value 4 is allowed to enter, since this is the abscissa of the given points. The remaining numeric values are entered, taking into account which value is selected in \square^2 .

If entered incorrectly, an error reporting with the correct answer appears.

As a result, there should be

$$\frac{\begin{array}{|c|c|c|} \hline x & - & 3 \\ \hline \end{array}}{\begin{array}{|c|c|c|} \hline 4 & - & 3 \\ \hline \end{array}} = \frac{\begin{array}{|c|c|c|} \hline y & - & 2 \\ \hline \end{array}}{\begin{array}{|c|c|c|} \hline -1 & - & 2 \\ \hline \end{array}} = \frac{\begin{array}{|c|c|c|} \hline z & - & -1 \\ \hline \end{array}}{\begin{array}{|c|c|c|} \hline 3 & - & -1 \\ \hline \end{array}}, \text{ or}$$

$$\frac{\begin{array}{|c|c|c|} \hline x & - & 4 \\ \hline \end{array}}{\begin{array}{|c|c|c|} \hline 3 & - & 4 \\ \hline \end{array}} = \frac{\begin{array}{|c|c|c|} \hline y & - & -1 \\ \hline \end{array}}{\begin{array}{|c|c|c|} \hline 2 & - & -1 \\ \hline \end{array}} = \frac{\begin{array}{|c|c|c|} \hline z & - & 3 \\ \hline \end{array}}{\begin{array}{|c|c|c|} \hline -1 & - & 3 \\ \hline \end{array}}$$

This thesis is concerned with the problem of efficiently analyzing large models of computer science obtaining the exact solution of such problems is often not possible to do easy. Approximation techniques have been proposed that importantly reduce the cost of computer analysis; however, these techniques lack bounds on errors that they introduce. The thesis proposes algorithms that bound rather than only estimates the of computer science solution.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ В МЕТОДІ АНАЛІЗУ
ІЄРАРХІЙ ДЛЯ СИСТЕМНОГО АНАЛІЗУ ПРИ ВИБОРІ
АЛЬТЕРНАТИВНИХ РІШЕНЬ В «ПП ТКАЧЕНКО А.С.»**

*Д.С. Рудяга, студент гр. І-41, спеціальності «Інформатика»
Вищий навчальний заклад Укоопспілки «Полтавський
університет економіки і торгівлі»
drudiaha@gmail.com*

*О.О. Ємець, д.ф.-м.н., професор
Вищий навчальний заклад Укоопспілки «Полтавський
університет економіки і торгівлі»
yemetsli@ukr.net*

Ставиться задача системного аналізу шляхом застосування методу аналізу ієрархій вибору альтернативних рішень в «ПП Ткаченко О.С.». Розробити програмне забезпечення однієї задачі системного аналізу роботи «ПП Ткаченко О.С.».

Rudiaha D.S., Iemets O.O. Software in the hierarchy analysis method for system analysis when choosing alternative solutions in the enterprise "Tkachenko AS"

The task of the system analysis by applying the method of analysis of hierarchies of choice of alternative solutions in the enterprise "Tkachenko A.S." , providing the software of one task of system analysis of Tkachenko A.S.

Ключові слова: Аналіз, системний аналіз, аналіз ієрархій, програмне забезпечення.

Keywords: Analysis, system analysis, analysis of hierarchies, software.

Метод аналізу ієрархій – це системна процедура, розроблена відомим американським математиком Т.Л. Сааті в

1977 році [1-5], і оснований на наступних принципах:

- принцип ідентичності і декомпозиції;
- принцип дискримінації і порівняльних суджень;
- принцип синтезу.

Реалізація принципу ідентичності і декомпозиції здійснюється на першому етапі застосування МАІ, в якому передбачається структурування проблеми у вигляді ієрархії. Такий підхід до розгляду складної задачі дозволяє уникнути складних порівнянь, замінивши їх попарними, і крім того має засоби для перевірки послідовності (несуперечливості) тверджень експерта. Цим пояснюється широка розповсюдженість методу, його дійсно системний характер та велика кількість практичних застосувань.

У найпростішому випадку ієрархія будується з вершини (цілі - з точки зору управління), через проміжні рівні (критерії, від яких залежать наступні рівні) до найнижчого рівня (який найчастіше є переліком альтернатив). Тобто при розв'язанні задачі аналітичного планування принцип ідентичності і декомпозиції буде застосовано рівно стільки разів скільки ієрархій прямих та зворотних процесів планування буде побудовано.

Звичайно, слід відзначити, що, як і у всіх методів системного аналізу, не існує єдиного шляху, що приводить до істини, але за умови залучення кваліфікованих експертів та досвідченого системного аналітика метод аналізу ієрархій дозволяє генерувати та відшукувати рішення, що відповідатимуть призначенню складної системи.

Принцип дискримінації і порівняльних суджень реалізується на другому етапі методу аналізу ієрархій і його суть полягає в застосуванні методу попарних порівнянь, зміст якого полягає в наступному. Нехай задана деяка фіксована множина об'єктів, які порівнюються з точки зору їх переваги, бажаності та важливості. Результати записуються у вигляді матриць попарних порівнянь, елементами яких є числа, що виражають важливість (або відносний вплив) кожних двох складових проблеми відносно елемента, що примикає з верхнього рівня ієрархії. Результат порівняння відображає не

лише факт, але і степінь (силу, інтенсивність) переваги, а в задачах динамічного планування й залежність переваг від часу. Тому елементами матриць попарних порівнянь для динамічних задач можуть бути не лише константи, а й функції залежності від часу. Для проведення суб'єктивних парних порівнянь елементів розроблено спеціальну шкалу відносної важливості, яку наведено в таблиці 1.1.

Таблиця 1.1 – Шкала відносної важливості

Інтенсивність	Означення	Пояснення
1	Рівнозначна важливість	Рівний внесок двох видів діяльності в досягнення мети
3	Поміrkована перевага одного над іншим	Досвід та переконаність дають суттєву перевагу одного виду діяльності над іншим
5	Суттєва або поміrkована значна перевага	Досвід та переконання дають значну перевагу одного виду діяльності над іншим
7	Значна перевага	Одному виду діяльності надається настільки значна перевага, що вона є практично переважною
9	Занадто значна перевага	Перевага одного виду діяльності над іншим не викликає сумніву
2, 4, 6, 8	Проміжні значення між двома сусідніми переконаннями	Застосовується при компромісних випадках
Обернені величини наведених чисел	Якщо при порівнянні одного виду діяльності з іншим отримано одне з вищезазначених чисел, то при порівнянні іншого виду діяльності з першим отримаємо обернене значення	

Розглянута шкала відносної важливості має і декілька недоліків. Оскільки, переваги експерта, на основі яких будуються матриці попарних порівнянь та обирається

найкраща альтернатива, є „прив'язаними” до певного моменту часу, і результати такого вибору можуть застосовуватися в майбутньому лише за умови стаціонарності середовища. В більшості випадків переваги експерта мають тенденцію до зміни в часі, тобто бажано знати, якими будуть переваги експерта в той чи інший момент часу.

У рядках та у стовпцях матриць попарних порівнянь записують назви компонентів, які необхідно порівняти. При заповненні матриці варто починати з лівого елемента і ставити запитання: на скільки він є важливішим за елемент вгорі? У процесі реалізації попарних порівнянь доцільно запитання формулювати ступним чином:

- який з двох варіантів важливіший чи більше впливає?
- який з двох варіантів ймовірніший?
- який з двох варіантів має більшу перевагу?

Для більшості застосувань запитання потрапляє, зазвичай, в одну з цих категорій. При порівнянні критеріїв зазвичай питають, який з критеріїв є важливішим; при порівнянні альтернатив за критеріями – яка з альтернатив є бажанішою; при порівнянні сценаріїв, які отримують з критеріїв – який зі сценаріїв є більш ймовірним.

При порівнянні елемента із самим собою відношення дорівнює одиниці. Якщо перший елемент є важливішим за другий, то використовується обране число із шкали відносної важливості (див. таблицю 1.1), у протилежному випадку використовується обернена величина. Таким чином, зворотні один одному відношення заносяться в симетричні позиції матриці, тобто матриця попарних порівнянь є обернено симетричною, і для всіх її елементів має виконуватися рівність:

$$a_{ij} = \frac{1}{a_{ji}}$$

а всі елементи головної діагоналі такої матриці є одиницями.

Матриці попарних порівнянь містять однакову кількість рядків і стовпців, а також інші корисні характеристики, такі,

як власні вектори (V_i) і власні значення (λ_i), вектори локальних пріоритетів (P_i), індекс узгодженості (IY) та відношення узгодженості (BI). Існує велика кількість різних алгоритмів для знаходження цих характеристик квадратної матриці, але найчастіше для їх пошуку використовують наступні формули:

$$V_i = \sqrt[n]{\prod_{j=1}^n a_{ij}}$$

$$P_i = \frac{V_i}{\sum_{j=1}^n V_j}$$

Для динамічних задач матриця парних порівнянь містить функції залежності від часу у якості елементів, тому максимальне власне значення λ_{\max} а також власний вектор V_i також будуть залежати від часу, тобто

$$A(t)V_i(t) = \lambda_{\max}(t)V_i(t),$$

де $A(t)$ – матриця попарних порівнянь об'єктів, що містить інформацію про зміну переваги однієї альтернативи над іншою на деякому проміжку часу.

У випадках невеликого розміру матриці попарних порівнянь та простої функції, що апроксимує динаміку переваг експерта, розв'язання рівняння (1.4) можна отримати в аналітичному вигляді, в більшості ж випадків розв'язок отримується за допомогою чисельних методів. Окрім того, можливо створення системи, що накопичує інформацію про реальні переваги експерта в минулих періодах та шляхом екстраполяції прогнозує їх на близьке майбутнє.

У процесі формування матриці попарних порівнянь на матрицю накладається обмеження оберненої симетричності, тобто за умовою (1.3), що сприяє поліпшенню однорідності та послідовності тверджень експерта, тобто в числових твердженнях якщо один елемент в m разів переважає інший, то останній в $1/m$ разів переважає перший (або в m разів гірший). У практичних задачах кількісна (кардинальна) і транзитивна

(порядкова) однорідність (узгодженість) порушується, оскільки експерт оцінює переваги шляхом попарних порівнянь, а тому рівність $a_{ij}a_{jk} = a_{ik}$, що повинна б була виконуватися для всіх i, j, k , буде порушуватися. Чим більші порушення цих рівностей, тим меншою мірою можна довіряти результатам опитування експерта, і це свідчитиме, насамперед, про суперечливість тверджень експерта або ж (як один з виявів цього й є суперечливість) про некомпетентність в даній предметній області.

При порушенні однорідності ранг матриці попарних порівнянь відмінний від одиниці і вона буде мати декілька власних значень, а умова оберненої симетричності забезпечить невід'ємність всіх компонент головного власного вектора. Однак при невеликих відхиленнях тверджень від однорідності одне з власних значень буде істотно більше за інші і приблизно дорівнюватиме порядку матриці. Отже, для оцінки однорідності тверджень експерта доцільно використати відхилення величини максимального власного значення λ_{\max} від порядку матриці n .

Отримана в результаті опитування експерта матриця буде неузгодженою, тобто відображати певну непослідовність тверджень експерта, яка в реальних умовах неявна завжди. Корисним результатом для оцінювання неузгодженості є індекс узгодженості, який дає інформацію про ступінь порушення числової та транзитивної - порядкової узгодженості. Якщо відхилення від узгодженості перевищують межі, то доцільно їх перевірити в матриці. Індекс узгодженості розраховуємо за формулою:

$$IY = \frac{\lambda_{\max} - n}{n - 1}$$

де

$$\lambda_{\max} = \frac{1}{n} \sum_{j=1}^n \lambda_j$$

$$\lambda_j = \frac{\sum_{j=1}^n a_{ij}}{V_i}$$

Обчислений індекс узгодженості порівнюємо зі значенням, яке отримується за умови випадкового вибору кількісних значень з шкали відносної важливості зі збереженням умови оберненої симетричності випадкової матриці.

Ці значення отримані з використанням генератора псевдовипадкових чисел програмним шляхом: генерується випадкова матриця заданого розміру n , в якій діагональ заповнюється одиницями, а верхня трикутна частина - числами з відносної шкали важливості, що розподілені рівномірно з ймовірністю виникнення кожного значення,

$\frac{1}{n}$

рівною n . Нижня трикутна складова заповнюється, виходячи зі співвідношення (1.3), що визначає обернену симетричну матрицю. Для з генерованої таким чином матриці розраховується значення індексу узгодженості тверджень експерта. Процедура повторюється велику кількість разів з розрахунком середнього значення, яке й вноситься до таблиці 1.2 [1-5].

Таблиця 1.2 – Значення індексу узгодженості для випадкових матриць

Розмір матриці (n)	1	2	3	4	5	6	7	8	9	10
Випадковий індекс (BI)	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49

Відношення узгодженості є часткою від ділення індексу узгодженості на відповідне значення випадкової узгодженості, тобто

$$BU = \frac{IU}{BI}.$$

Якщо отримане значення є меншим, ніж 10%, то рівень узгодженості може вважатися задовільним. У деяких випадках можна обмежитися 20%.

При практичній реалізації методу аналізу ієрархій необхідно враховувати, що для обґрунтованих числових порівнянь оптимальною кількістю є 7 ± 2 елементи. Якщо розглядається більша кількість об'єктів, то необхідно застосовувати ієрархічну декомпозицію. У цьому випадку елементи групуються таким чином, щоб у групі було приблизно сім елементів. Елемент з найбільшою вагою в класі включається також в наступний клас елементів з більшими вагами і забезпечує однорідність шкали. В інших випадках можливо введення над критеріїв зі встановленням важливості під критеріїв відносно над критеріїв.

Принцип синтезу в методі аналізу ієрархій використовується для зважування локальних пріоритетів вагами критеріїв та знаходження глобальних пріоритетів. Розрахунок глобальних пріоритетів, тобто пріоритетів альтернатив відносно всієї ієрархії, є основним завданням методу аналізу ієрархій. Вихідними даними при цьому є результати опитування експертів у вигляді матриць попарних порівнянь при всіх вузлах ієрархії за винятком рівня альтернатив (контрастних сценаріїв). Щоб виявити глобальні пріоритети альтернатив необхідно скласти ще одну матрицю G , елементами якої є вектори локальних пріоритетів кожної з альтернатив, розташовані відповідно до послідовності критеріїв. До кожного стовпця векторів цієї матриці вгорі дописується пріоритет відповідного критерію.

Пріоритети синтезують, починаючи з рівня альтернатив. Локальні пріоритети перемножують на пріоритети відповідних критеріїв на вищому рівні і знаходять суму добутків відповідних рядків. Тобто, елементами вектора глобальних пріоритетів є скалярні добутки першого рядка матриці G з кожним із нижчих рядків. Цю процедуру повторюють, доки не

буде досягнуто рівня мети. Наприкінці одержують глобальний вектор пріоритетів альтернатив відносно цілі.

Розрахунок узагальненої міри узгодженості (для всієї ієрархії) здійснюється як добуток (скалярний) вектору індексів узгодженості 2-го рівня на вектор пріоритетів 1-го рівня.

Узагальнений індекс узгодженості M рахуємо як суму індексів узгодженості 1-го та 2-го рівнів. Аналогічно розраховуються сумарний випадковий індекс.

Відношення узгодженості для всієї ієрархії обчислюється за формулою M/\tilde{M} . Якщо $VU < 0,2$ ступінь узгодженості прийнятна.

Список використаних джерел

1. Саати Т. Аналитическое планирование. Организация систем / Т. Саати, К. Кернс ; [пер. с англ.]. – М.: Радио и связь, 1991 – 224 с.

2. Рогоза М.Є. Системи підтримки прийняття рішень: навч. посіб. / М.Є. Рогоза, О.О. Ємець, Є.М. Ємець. – Полтава: ПУЕТ, 2013. – 328 с.

3. Андрейчиков А.В. Анализ, синтез, планирование решений в экономике / А.В. Андрейчиков., О.Н. Андрейчикова. – М.: Финансы и статистика, 2001 – 368 с.

4. Лямец В.И. Системный анализ. Вводный курс / В.И. Лямец, А.Д. Тевяшев. – Харьков: ХНУРЕ, 2004. – 448 с.

5. Ємець О.О. Системний аналіз інноваційної діяльності на підприємствах газовидобувної галузі України / О.О. Ємець, В.П. Світалка. – К.: Наук. думка, 2008. – 202 с.

УДК 004

**АЛГОРИТМ РОБОТИ ТРЕНАЖЕРУ З ТЕМИ «НАВЧАННЯ
ЕЛЕМЕНТАРНОГО ПЕРСЕПТРОНУ» ДИСЦИПЛІНИ
«НЕЙРОННО-МЕРЕЖЕВІ ТЕХНОЛОГІЇ В
ІНФОРМАТИЦІ»**

В.В. Кильник, студент групи КН-51
Полтавський університет економіки і торгівлі
Puet12vadim@gmail.com

Ю. Ф. Олексійчук, к. ф.-м. н.
Полтавський університет економіки і торгівлі
olexijchuk@gmail.com

*У публікації розглядається алгоритм тренажеру з теми
«Навчання елементарного персеプトрону»*

*Kylnyk V. V., Oleksiichuk Yu. F. The algorithm of simulator's on
the topic "Teaching elementary perceptron" for the discipline
"Neural networks in the field of informatics". The publication
considers the implementation of the algorithm simulator's for the
topic "Teaching elementary perceptron"*

Ключові слова: ДИСТАНЦІЙНЕ НАВЧАННЯ,
СИМУЛЯТОР, ПЕРСЕПТРОН, НЕЙРОННІ МЕРЕЖІ

Keywords: DISTANCE LEARNING, SIMULATOR,
PERCEPTRON, NEURAL NETWORK

В зв'язку з впровадженням дистанційної форми навчання розробка навчальних тренажерів є актуальною задачею [1-5].

Завданням роботи є алгоритму тренажеру за темою «Навчання елементарного персеプトрону» з дисципліни «Нейронно-мережеві технології в інформатиці». Навчальним тренажером називаємо інформаційно-довідкову та тренувальну систему, а також систему навчання теоретичним і практичним

основам навчання елементарного персеPTRону [6] та розрахунку його коефіцієнтів, що використовує різноманітні способи засвоєння знань та їх контролю.

Виконання поставленої задачі було розпочато з розробки алгоритму майбутнього тренажеру. В алгоритмі потрібно передбачити основні кроки роботи програмного продукту. При розробці було враховані такі можливості тренажеру як: перегляд інформації про розробника, компактний теоретичний матеріал за темою тренажеру, різноманітні способи взаємодії з користувачем з використанням підказок та різних типів варіантів роботи з інформацією.

На першому етапі роботи з тренінгом користувач у тестовому режимі відповідає на ряд теоретичних питань із теми, що дає можливість перевірити рівень підготовленості користувача з даної теми.

На наступному етапі користувачеві відображається схематичне зображення персеPTRону, та пропонується розрахувати його коефіцієнти для його тренування. Після натиснення кнопки отримання результату, програма виконує аналіз введених даних і якщо вони правильні перехід на наступний екран де зображений більш складний варіант з більшою кількістю коефіцієнтів. Якщо користувач помилився - пропонується спробувати повторно чи скористатися допомогою.

Після проходження трьох етапів різної складності користувач матиме змогу переглянути результати свого тренування і загальний бал. За бажанням, він може повторити тренування чи завершити роботу з тренажером.

Таким чином алгоритм тренажеру має такі кроки:

1. Завантаження тренажеру. Відображення автора. Можливість почати тренування чи перейти до теоретичних відомостей. Якщо користувач обирає «Розпочати тренування» перехід на наступний крок.
2. Відображення теоретичного запитання, варіантів відповідей. Якщо користувач натискає «Далі», збереження відповідей в пам'яті, перехід на наступний крок.
3. Після проходження теоретичних запитань, відображення

практичного завдання.

Якщо користувач натискає «Далі», виконуємо перевірку заповнених коефіцієнтів, при помилці пропонуємо користувачеві повторити чи скористатися допомогою.

Якщо помилок не виявлено тоді перехід на крок 4.

Якщо користувач обирає «Повторити» то відновлюємо форму, перехід на крок 3.

Якщо обирає «Допомога» тоді відображаємо інформацію яким чином потрібно заповнювати коефіцієнти. Перехід на крок 3.

4. Після завершення всіх кроків відображаємо інформацію про проходження з демонстрацією набраних балів та кількості помилок.

Якщо користувач обирає «Повторити тренування», тоді перехід на крок 2.

Якщо обрано «Завершити» - завершуємо роботу тренажеру.

Реалізацію тренажеру виконуємо за допомогою таких засобів як: мова програмування JavaScript, яка потрібна для створення сценаріїв роботи тренажеру і керування елементами сторінок, мова розмітки HTML5 для створення сторінок, каскадні таблиці стилів CSS3 для стилізування різних частин тренажеру. Для зручної та ефективної роботи, використовуємо інтегроване середовище розробки JetBrains WebStorm. Обрані технології дозволяють створювати мультиплатформні веб-застосунки, які можливо використовувати на великій кількості пристроїв, що мають вбудований чи інсталюваний веб-браузер, таких як: мобільні телефони і планшети на базі IOS чи Android, комп'ютери і ноутбуки з різноманітними операційними системами. Обрані технології є більш гнучкими за Java чи C# тому, що в першу чергу не потребують встановлення додаткового програмного забезпечення.

У доповіді продемонстровано алгоритм тренажеру, за допомогою якого користувач може отримати практичні навички, краще зрозуміти принципи роботи елементарного перцептону та перевірити свої знання за темою «Навчання елементарного

персептрону» дисципліни «Нейронно-мережеві технології в інформатиці». Розглянуто технології реалізації тренажеру.

Література

1. Ємець О. О. Про розробку тренажерів для дистанційних курсів кафедрою ММСІ ПУЕТ / О.О. Ємець // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.). – Полтава: ПУЕТ, 2015. – С. 152-161.
2. Ольховська О. В. Технології підтримки системи дистанційного навчання в Полтавському університеті економіки і торгівлі / О. В. Ольховська, Д. М. Ольховський // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – С. 219-221.
3. Кильник В. В. Алгоритм тренажера з теми «Сортування методом перемішування» дистанційного курсу «Алгоритми та структури даних» / В. В. Кильник // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю (м. Полтава, 16–18 березня 2017 р.) – Полтава: ПУЕТ, 2017. – С. 135-137.
4. Русін В. С. Програмна реалізація елементів тренажеру з теми "Аналіз алгоритму сортування вставками" дисципліни "Аналіз алгоритмів" / В. С. Русін, Ю. Ф. Олексійчук // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю (м. Полтава, 16–18 березня 2017 р.) – Полтава: ПУЕТ, 2017. – С. 236-237.
5. Олексійчук Ю. Ф. Розробка та впровадження дистанційного курсу з дисципліни «Програмування» / Ю. Ф. Олексійчук // Дистанційна освіта: забезпечення доступності та неперервної освіти впродовж життя (e-learning and university education-2017): матеріали XLII Міжнародної науково-методичної

- конференції (м. Полтава, 9–10 лютого 2017 року) – Полтава: ПУЕТ, 2017. – С. 167-169.
6. Хайкин С. Нейронные сети: полный курс. / С. Хайкин – М.: ООО "И. Д. Вильямс", 2006. – 1104 с.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ЩО РЕАЛІЗУЄ МЕТОД БАЖАНОЇ ТОЧКИ В ПРИЙНЯТТІ РІШЕНЬ

А. І. Доба, студент групи КН-51

О.О. Ємець, д. ф.-м. н., професор

*Полтавський університет економіки і торгівлі
marclas255@gmail.com*

Розглядається метод бажаної точки та його програмна реалізація.

Doba A. I., Iemets O.O. Software implementing the desired point method for decision-making. The method of the desired point and its program realization are considered.

Ключові слова: ПРИЙНЯТТЯ РІШЕНЬ, БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ, МЕТОД БАЖАНОЇ ТОЧКИ

Keywords: DECISION MAKING, MULTICRITERIA OPTIMIZATION, DESIRED POINT METHOD.

В доповіді розглядається один з методів багатокритеріальної оптимізації [1] – метод бажаної точки. Розглянемо його ідею [2].

Це діалогова процедура, особливістю якої є необхідність задання особі, що приймає рішення (ОПР), бажаних значень критеріїв для визначення переваги на множині критеріїв.

0-ий крок. Розраховуються «найкращі» та «найгірші» значення критеріїв: $f_i^* = \max_{x \in X} f_i(x)$, $h_i^* = \min_{x \in X} f_i(x)$, $i = \overline{1, m}$.

Здійснюється монотонне перетворення критеріїв до нормованого безрозмірного вигляду: $w_i(x) = \frac{f_i^* - f_i(x)}{f_i^* - h_i^*}$,

$i = \overline{1, m}$.

к-ий крок ($k = 1, 2, \dots$). ОПР аналізує отриманий на попередньому кроці розв'язок та його оцінку у порівнянні з «найкращими» і «найгіршими» значеннями критеріїв і вказує

бажані значення критеріїв $\xi_i^k \in [h_i^*, f_i^*]$, $i = \overline{1, m}$. Здійснюються перетворення бажаних значень цільових функцій до нормованого безрозмірного вигляду $w_i^k(x) = \frac{f_i^* - \xi_i^k}{f_i^* - h_i^*}$, $i = \overline{1, m}$.

Обчислюються вагові коефіцієнти критеріїв:

$$\rho_i^k = \prod_{j=1, j \neq i}^m w_j^k / \sum_{j=1}^m \prod_{l=1, l \neq j}^m w_l^k, \quad i = \overline{1, m}.$$

Ефективна альтернатива x^k знаходиться як розв'язок однокритеріальної задачі $\max_{x \in X} \min_{i=1, m} \rho_i^k w_i(x)$. Обчислюється

оцінка $y^k = (f_1(x^k), \dots, f_m(x^k))$. Якщо отримані значення цільових функцій задовольняють ОПР, то процедура закінчується, у протилежному випадку переходимо на наступний крок.

Цей метод використовує тільки один тип інформації від ОПР про бажані значення критеріїв.

Була здійснена алгоритмізація методу. Метод бажаної точки запрограмовано з використанням середовища візуального програмування *Delphi* та мови програмування *Object Pascal*. Оскільки, метод бажаної точки передбачав використання градієнтного методу та методу Монте-Карло, ці методології також були запрограмовані. Всі проміжні розрахунки, зокрема, і для градієнтного методу виводяться у файл.

Література

1. Ємець О. О. Методи оптимізації та дослідження операцій [Електронний ресурс]: навчально-методичний посібник за кредитно-модульною організацією навчального процесу / О. О. Ємець, Т. О. Парфьонова. – Полтава: ПУЕТ, 2013. – Режим доступу: локальна мережа ПУЕТ.
2. Волошин О. Ф. Теорія прийняття рішень / О. Ф. Волошин, С. О. Машченко. – К.: Видавничо-поліграфічний центр «Київський університет», 2006. – 304 с.

**РОЗРОБКА ЕЛЕМЕНТІВ ТРЕНАЖЕРУ ЗА ТЕМОЮ
«ПАРАМЕТРИ ЛІНІЙНОЇ РЕГРЕСІЇ» З ДИСЦИПЛІНИ
«ОБЧИСЛЮВАЛЬНІ МЕТОДИ»**

А. В. Задорожний, студент КН-51

Т. В. Чілікіна к. ф.-м. н., доц.

Полтавський університет економіки і торгівлі

Zador1278@gmail.com

*У публікації розглядається алгоритм тренажеру з теми
«Параметри лінійної регресії»*

*Zadorozhnyy A.V., Chilikina T. V. The development of elements of
the simulator on the topic “Parameters of linear regression” for the
discipline “Calculate methods”. The publication considers the
implementation of the algorithm of simulator for the topic
“Parameters of linear regression”*

Ключові слова: ЛІНІЙНА РЕГРЕСІЯ, ДИСТАНЦІЙНЕ
НАВЧАННЯ, ТРЕНАЖЕР.

Keywords: LINEAR REGRESSION, DISTANCE LEARNING,
SIMULATOR.

Завданням роботи є розробка елементів тренажеру за темою «Параметри лінійної регресії» з дисципліни «Обчислювальні методи». Навчальним тренажером будемо називати інформаційно-довідкову систему, а також систему навчання теоретичним і практичним основам, яка застосовує різні види засвоєння та контролю знань.

При виконанні поставленого завдання необхідно в першу чергу розробити алгоритм роботи тренажера. При запуску програмного продукту користувач має можливість переглянути теоретичний матеріал з теми, інформацію про розробника, а також для зручності роботи із поставленим завданням, можливість завантажити Microsoft Office Excel.

На першому етапі роботи з тренінгом користувач у тестовому режимі відповідає на ряд теоретичних питань із теми, що дає

можливість перевірити рівень підготовленості користувача за даної теми.

На другому етапі користувачу відображується робоче вікно тренажера: умова завдання, а саме: «За заданою вибіркою (табл.1) провести аналіз залежності витрат на відпустку Y (тис.у.о.) домогосподарства від кількості членів родини X , результати якої наведено у таблиці (табл. 1). Визначити вид залежності, оцінити параметри рівняння регресії, оцінити силу лінійної залежності між X та Y ».

Таблиця 1. – Задана вибірка

<i>№ n/n</i>	<i>Кількість членів родини</i> x_i	<i>Витрати на відпустку (тис. у.о.)</i> y_i	$x_i y_i$	x_i^2	y_i^2
1	2	3	4	5	6
1	1	16			
2	2	12			
3	5	23			
4	4	19			
5	6	30			
Всього Σ					
$n =$					
$b1 =$					
$b0 =$					

На наступних етапах роботи тренажера необхідно відповідно до заданої вибірки поступово заповнити таблицю, розрахувавши значення параметрів послідовно.

Так на третьому етапі визначаємо визначення значення параметру n – обсягу вибірки. Якщо користувач вірно вводить значення в активну комірку, то відбувається перехід до наступного кроку, а якщо допускає помилку, то виводиться повідомлення-підказка, наприклад: «Обсяг вибірки рівний

кількості її елементів, тобто 5».

На четвертому етапі визначаємо значення стовпців $x_i y_i$, x_i^2 та суми по стовпцям 2-4 таблиці.

На п'ятому етапі обчислити параметри регресії b_1 та b_0 та значення стовпця **ур**. Після завершення розрахунків як перевірка виконаних розрахунків передбачається можливість побудови рівняння отриманої лінійної регресії і кореляційного поля за вихідними даними.

Закінчення роботи тренажера характеризується записом «Тренінг закінчено».

На основі отриманих результатів користувач повинен провести аналіз виду залежності регресії, оцінити параметри рівняння регресії лінійної залежності між фактором x та показником y .

Реалізацію тренажеру передбачається виконати за допомогою мови програмування високого рівня JavaScript.

У доповіді розглянуто алгоритм реалізації тренажера, за темою “Параметри лінійної регресії” з дисципліни «Обчислювальні методи».

Література

1. Ємець О. О. Про розробку тренажерів для дистанційних курсів кафедри ММСІ ПУЕТ / О.О. Ємець // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.). – Полтава: ПУЕТ, 2015. – С. 152-161.
2. Задорожній А.В. Створення програмного забезпечення тренажеру за темою "Наближені методи розв'язування СЛАР" з дисципліни "Обчислювальні методи" / А.В.Задорожній // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю (м. Полтава, 16–18 березня 2017 р.) – Полтава: ПУЕТ, 2017. – С. 118-121.

Алфавітний покажчик авторів

Бах Абібу	40
Безменов Е. Г.	9
Голубенко В.О.	11
Доба А.І.	59
Дяченко В.О.	17
Ємець О.О.	4, 21, 31, 45, 59
Задорожній А.В.	61
Кильник В.В.	54
Куркін В.В.	21
Мандрика В.М.	27
Олексійчук Ю.Ф.	11, 27, 54
Парфьонова Т.О.	40
Рудяга Д.С.	45
Сивокінь О.Ю.	4
Стовбун Д.О.	31
Чілікіна Т.В.	61
Щербак О.В.	37